```
function llk = liklWeitz_crude_2(param,dat,D,nalt,epsilonDraw,etaDraw)

%data features
consumer=dat(:,1);
N_obs=length(consumer);
N_cons=length(unique(consumer));

%choices
tran=dat(:,end);
searched=dat(:,end-1);
last=dat(:,end-4);
has_searched=dat(:,end-3);

%parameters
outside=dat(:,3);
c=exp(param(end)).*ones(N_obs,1);
X=dat(:,4:3+size(param(1:end-1),2));
xb=sum(X.*param(1:end-1),2);
eut=(repmat(xb,1,D)+etaDraw).*(1-outside);
ut=eut+epsilonDraw;

%%%%%%FORM Z's%%%%%%%%%
%%%1. look-up table method
table=importdata('tableZ.csv');
m=zeros(N_obs,1);
for i=1:N_obs
    lookupvalue=abs(table(:,2)-c(i));
    if (table(1,2)>=c(i)&& c(i)>=table(end,2))
        [~,index_m]=min(lookupvalue);
        m(i)=table(index_m,1);
    elseif table(1,2)<c(i)
        m(i)=-c(i);
    elseif c(i)<table(end,2)
        m(i)=4.001;
    end
end
z=m+eut;


% %%%2. newton method
% m=zeros(N_obs,1);
% x0 = 0; % initial point
% for i = 1:size(c, 1)
%     m(i) = newtonZ(c(i), x0);
% end
% z= eut + m;

% %%%3. contraction mapping method
% m=zeros(N_obs,1); % initial point
% for i = 1:size(c, 1)
```

```
%     m(i) = contractionZ(m(i), c(i));
% end
% z = eut + m;


%%%%%%%%%%%%%%%%%%%%
ut_searched=ut;
searched2=repmat(searched,1,D);
ut_searched(searched2==0)=NaN;

for d=1:D
    %best ut_so_far
    ymax=cummax(reshape(ut_searched(:,d),nalt,N_cons));
    ymax=circshift(ymax,1);
    ymax=reshape(ymax,N_obs,1);
    %best z_next
    zmax=cummax(reshape(z(:,d),nalt,N_cons),'reverse');
    zmax=circshift(zmax,-1);
    zmax=reshape(zmax,N_obs,1);
    %outside option for each consumer
    u0_2=ut(:,d).*outside;
    u0_3=reshape(u0_2,nalt,N_cons);
    u0_4=repmat(sum(u0_3),nalt,1);
    u0_5=reshape(u0_4,N_obs,1);


    %selection rule: z>z_next
    supp_var = ones(size(dat,1),1);
    order=(z(:,d)-zmax).*has_searched.*searched.*(1-outside).*(1-last) +
supp_var.*last + supp_var.*outside + supp_var.*(1-has_searched) +
supp_var.*(1-searched);

    order = (order > 0);

    %stopping rule: z>u_so_far
    search_1 = (z(:,d)-ymax).*has_searched.*searched.*(1-outside) +
supp_var.*outside + supp_var.*(1-searched) + supp_var.*(1-has_searched);

    search_2 = (ymax-z(:,d)).*has_searched.*(1-searched) +
supp_var.*(1-has_searched) + supp_var.*searched;

    search_3 = (u0_5-z(:,d)).*(1-has_searched).*(1-outside) +
supp_var.*has_searched + supp_var.*outside;


    search_1 = (search_1 > 0);
    search_2 = (search_2 > 0);
    search_3 = (search_3 > 0);

    %choice rule
```

3 行 3 列の行列の列の累積最大値を求めます。

```
A = [3 5 2; 1 6 3; 7 8 1]
```

```
A = 3×3

     3     5     2
     1     6     3
     7     8     1
```

```
M = cummax(A)
```

```
M = 3×3

     3     5     2
     3     6     3
     7     8     3
```

```
A = (1:10)'
```

```
A = 10×1

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
```

circshift を使用して要素を 3 位置分シフトします。

```
Y = circshift(A,3)
```

```
Y = 10×1

     8
     9
    10
     1
     2
     3
     4
     5
     6
     7
```

The optimal decision rules described in Section 2.2.2 fully describe optimal search and purchase behavior. According to the *selection rule*, it must be that products are searched in decreasing order of reservation utilities:

$$z_{ih} \geq \max_{k \in \Im \setminus \{1,\ldots,h\}} z_{ik}, \qquad \forall h \in S_i. \tag{14}$$

In addition, the *stopping rule* imposes the following two restrictions: for the set of searched options, it must be that

$$z_{ih} \geq \max_{k=0}^{h-1} u_{ik}, \qquad \forall h \in S_i. \tag{15}$$

In contrast, for the options that were not searched, it must be that

$$\max_{h \in S_i \cup \{0\}} u_{ih} \geq \max_{l \in \bar{S}_i} z_{il}. \tag{16}$$

Finally, consistent with the *choice rule*, if the consumer chooses y_i, then her utility from this option is larger than that of any other searched product (including the outside option), i.e.,

$$u_{iy_i} \geq \max_{h \in S_i \cup \{0\}} u_{ih}. \tag{17}$$

$$\begin{aligned} L_i(\boldsymbol{\theta}) \;=\; & \underbrace{Pr(z_{ih} \geq \max_{k \in \Im \setminus \{1,\ldots,h\}} z_{ik} \; \forall h \in S_i}_{\text{selection rule}} \\ & \underbrace{\cap \; z_{ih} \geq \max_{k=0}^{h-1} u_{ik} \; \forall h \in S_i \; \cap \; \max_{h \in S_i \cup \{0\}} u_{ih} \geq \max_{l \in \bar{S}_i} z_{il}}_{\text{stopping rule}} \\ & \underbrace{\cap \; u_{iy_i} \geq \max_{h \in S_i \cup \{0\}} u_{ih}}_{\text{choice rule}}). \end{aligned}$$

The optimal decision rules described in Section 2.2.2 fully describe optimal search and purchase behavior. According to the *selection rule*, it must be that products are searched in decreasing order of reservation utilities:

$$z_{ih} \geq \max_{k \in \Im \setminus \{1,\ldots,h\}} z_{ik}, \qquad \forall h \in S_i. \tag{14}$$

In addition, the *stopping rule* imposes the following two restrictions: for the set of searched options, it must be that

$$z_{ih} \geq \max_{k=0}^{h-1} u_{ik}, \qquad \forall h \in S_i. \tag{15}$$

In contrast, for the options that were not searched, it must be that

$$\max_{h \in S_i \cup \{0\}} u_{ih} \geq \max_{l \in \bar{S}_i} z_{il}. \tag{16}$$

Finally, consistent with the *choice rule*, if the consumer chooses y_i, then her utility from this option is larger than that of any other searched product (including the outside option), i.e.,

$$u_{iy_i} \geq \max_{h \in S_i \cup \{0\}} u_{ih}. \tag{17}$$

```
    u_ch2=ut(:,d).*tran;
    u_ch3=reshape(u_ch2,nalt,N_cons);
    u_ch4=repmat(sum(u_ch3),nalt,1);
    u_ch5=reshape(u_ch4,N_obs,1);

    choice = (u_ch5-ut(:,d)).*(1-tran).*searched + supp_var.*tran + supp_var.*(1 -
searched);

    choice = (choice > 0);

    %1. combine all inputs
    chain_mult = order.*search_1.*search_2.*search_2.*search_3.*choice;

    %2. sum at the consumer level
    final_result = accumarray(consumer, chain_mult, [N_cons 1],@prod);
    %3. prob for that d
    prob(:,d)=final_result;
end

%4. avg across D
llk=mean(prob,2);

end
```

consumerでgroup_byして、chainmultのtotal productを作って、1000*1のデータを作る

例

B = accumarray(ind,data,sz,fun) は、ind で指定された data の各グループに関数 fun を適用します。@ 記号を使用して fun を指定するか (たとえば、@mean)、あるいは [] を指定して既定の @sum を使用します。

$$L_i(\boldsymbol{\theta}) = Pr(\underbrace{z_{ih} \geq \max_{k \in \Im \setminus \{1,\ldots,h\}} z_{ik} \; \forall h \in S_i}_{\text{selection rule}} \cap \underbrace{z_{ih} \geq \max_{k=0}^{h-1} u_{ik} \; \forall h \in S_i \cap \max_{h \in S_i \cup \{0\}} u_{ih} \geq \max_{l \in \bar{S}_i} z_{il}}_{\text{stopping rule}} \cap \underbrace{u_{iy_i} \geq \max_{h \in S_i \cup \{0\}} u_{ih}}_{\text{choice rule}}).$$